CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 16, 18-23, and 61-74.
- After this Amendment: Claims 16, 18-23, and 61-74.

Non-Elected, Canceled, or Withdrawn claims: None

Amended claims: 16, 18-19, 22-23, 61-63, and 68-74.

New claims: None

Claims:

1-15. (Canceled)

16. (Currently Amended) A process for verification of a client authentication request by a server which can decrease problems associated with sham authentication requests, the process comprising:

receiving, in the server, a client authentication request including client-specific data;

comparing the client specific data to data stored in a first cache memory coupled to the server to determine ~~that~~ whether the client specific data meet a first threshold of validity;

~~when~~ if comparing determines that the client specific data meet the first threshold of validity, proceeding with the authentication process; and

~~when~~ if comparing determines that the client specific data do not meet the first threshold of validity, then storing in a second cache memory a portion of the client specific data ~~in a second cache memory along with~~ and an indication that the client specific data do not correspond to a valid client, wherein the portion of the client specific data and the indication stored in [[a]] the second cache memory ~~identifying~~ identify a client name associated with the client authentication request and ~~associating~~ associate the client name with a ~~valid~~ negative indication of validity regardless of whether the client specific data ~~included~~ includes valid proof of knowledge of privileged data, and then terminating the verification process.

17. (Canceled)

18. (Currently Amended) The process of claim 16, wherein:

proceeding with the authentication process comprises ~~second~~ comparing the client specific data with data stored in [[a]] the second cache memory to determine ~~when~~ whether the client specific data meet a second threshold of validity and ~~when~~ whether the client specific data correspond to an identity previously determined to be valid or invalid; ~~and~~

~~when~~ if the client specific data meet the second threshold, transmitting a request for verification to a database containing client specific data; and

~~when~~ if the client specific data correspond to an identity previously determined to be invalid, terminating the authentication request.

19. (Currently Amended) The process of claim 16, wherein receiving comprises receiving data[[.]] including one or more of: a name, a NameHash, a truncation of a NameHash, a NameKeyHash, a truncation of a NameKeyHash, a TimedNameKeyHash, a truncation of a TimedNameKeyHash or a time.

20. (Original) The process of claim 16, wherein receiving comprises receiving a TimedNameKeyHash.

21. (Original) The process of claim 16, wherein receiving comprises receiving a TimedNameKeyHash and a current time.

22. (Currently Amended) The process of claim 16, wherein comparing the client specific data to data stored in [[a]] the first cache memory comprises comparing a TimedNameKeyHash contained in the authentication request to a function of a stored NameKeyHash and a current time.

23. (Currently Amended) The process of claim 16, wherein receiving client specific data includes receiving a current time, and further comprising determining ~~when~~ whether the received current time disagrees with another current time used by the authentication server, and, when the received current time and the another current time disagree, sending the another current time to an originator of the authentication request.

24-60. (Canceled)

61. (Currently Amended) A computer system comprising: an authentication server; and

a ~~primary~~ first cache memory coupled to the authentication server, wherein the authentication server is configured to perform a method, the method comprising:

~~receive~~ receiving a client authentication request including client-specific data;

~~compare~~ comparing the client specific data to data stored in [[a]] the first cache memory coupled to the authentication server to determine ~~that~~ whether the client specific data meet a first threshold of validity;

~~when comparing determines that the client specific data meet the first threshold of validity, proceed with authentication; and~~

~~when~~ if comparing determines that the client specific data do not meet the first threshold of validity, terminate authentication and deny the authentication request;

~~second compare~~ if comparing determines that the client specific data meet the first threshold of validity, proceed with authentication by comparing the client specific data with data stored in [[the]] a second cache memory to determine ~~when~~ whether the client specific data meet a second threshold of validity and ~~when~~ whether the client specific data correspond to an identity previously determined to be valid or invalid;

~~when~~ if the client specific data meet the second threshold, ~~transmit~~ transmitting a request for verification to a database containing client-specific data; and

~~when~~ if the client specific data correspond to an identity previously determined to be invalid, ~~terminate~~ terminating the authentication request.

62. (Currently Amended) The computer system of [[.]] claim 61, wherein the authentication server is configured to employ a first, plaintext portion of the client-specific data as a ~~cachekey~~ cache key to obtain related encrypted client-specific data from the first cache memory.

63. (Currently Amended) The computer system of claim 61[[.]] wherein the authentication server is further configured to store [[.]] at least ~~some~~ a portion of the client specific data in [[a]] the second cache memory along with an indication that the client specific data do not correspond to a valid client ~~when~~ if comparing determines that the client specific data do not meet the first threshold.

64. (Previously Presented) The computer system of claim 61, wherein the client-specific data includes a NameKeyHash that is also a function of time.

65. (Previously Presented) The computer system of claim 61, wherein the client-specific data includes a TimedNameKeyHash.

66. (Previously Presented) The computer system of claim 61. wherein the client specific data includes a TimedNameKeyHash and a current time is included with the client-specific data.

67. (Previously Presented) The computer system of claim 61, wherein the client specific data stored in the first cache memory comprises a NameKeyHash, and wherein the authentication server is configured to form a TimedNameKeyHash from the NameKeyHash and to compare the formed TimedNameKeyHash to a portion of the client-specific data.

68. (Currently Amended) The computer system of claim 61[[,]] wherein the client specific data includes a current time, and wherein the authentication server is further configured to determine ~~when~~ whether the received current time disagrees with another current time, the another current time being used by the authentication server, and ~~when~~ if the received current time and the another current time disagree, send the another current time to an originator of the authentication request.

69. (Currently Amended) A process for verification of a client authentication request by a server which can decrease problems associated with sham authentication requests, the process comprising:

receiving, in the server, a client authentication request including client-specific data comprising a name[[.]] or hash of the name along with a client key or ~~some~~ a proof of knowledge which identifies the client key;

comparing the ~~client-specific~~ client-specific data to data stored in a first cache memory coupled to the server to determine ~~that~~ whether the client specific data meet a first threshold of validity,[[.]] wherein the first cache memory stores names and keys of valid clients, and wherein the first cache memory uses the name or the hash of the name as a ~~cashekey~~ cache key to access the first cache memory;

~~when~~ if comparing determines [[`]]that the ~~client-specific~~ client-specific data meet the first threshold of validity ~~since the name and the client key identified in the client authentication request corresponds to a valid entry in the first cache memory~~, the first threshold of validity being met when the name and the client key identified in the client authentication request correspond to a valid entry in the first cache memory, proceeding with the authentication process; and [[']]

~~when~~ if comparing determines that the ~~client-specific~~ client-specific data do not meet the first threshold of validity ~~since the name and the client key identified in the client authentication request does not correspond to a valid entry in the first cache memory~~, then storing the name ~~and the~~ , the client key, and validity/invalidity indicators in a second cache memory ~~along validity/invalidity indicators~~, wherein the name stored in the second cache memory is associated with a ~~valid~~ validity indication regardless of whether the client key or the proof of knowledge for the client key matches data in an associated authentication database, and ~~then~~ terminating the verification process.

70. (Currently Amended) A process for authenticating a user which can decrease problems associated with sham authentication requests, the process comprising:

receiving an authentication request including first client specific data comprising at least one of a client name and proof of knowledge of a client key;

computing a NameHash using the received client name and a random session key;

using data corresponding to the NameHash as a ~~cachekey~~ cache key to access first validity threshold data from a first cache memory;

comparing the first validity threshold data to the first client specific data; and

~~when~~ if comparing determines that the first client specific data do not meet the first threshold of validity, then storing a portion of the client specific data in a second cache memory along with an indication that the client specific data do not correspond to a valid client, the portion of the client specific data stored in [[a]] the second cache memory identifying a client name associated with the client authentication request and associating the client name with a ~~valid~~ validity indication regardless of whether the client specific data included valid proof of knowledge of privileged data, and then terminating the verification process.

71. (Currently Amended) The process of claim 70, further comprising, when the first validity data do not match the first client data, storing the client key and a CredentialInvalidFlag in [[a]] the second cache memory.

72. (Currently Amended) The process of claim 70, further comprising, when the first validity data do match the first client data, employing the client name as a ~~cachekey~~ cache key to access second client validity data from [[a]] the second cache memory.

73. (Currently Amended) The process of claim 70, further comprising, when the first validity data do match the first client data, employing the client name as a ~~cachekey~~ cache key to access second client validity data from [[a]] the second cache memory, wherein the second client validity data comprise a stored copy of a client key.

74. (Currently Amended) The process of claim 70, wherein using data corresponding to the NameHash as a ~~cachekey~~ cache key comprises using a truncation of the NameHash to access first validity threshold data from [[a]] the first cache memory.